



Parallel CE/SE Computations via Domain Decomposition

Ananda Himansu
Taitech Inc., Cleveland, Ohio

Philip C.E. Jorgenson
Glenn Research Center, Cleveland, Ohio

Xiao-Yen Wang
Taitech Inc., Cleveland, Ohio

Sin-Chung Chang
Glenn Research Center, Cleveland, Ohio

Prepared for the
First International Conference on Computational Fluid Dynamics
sponsored by the Kyoto Institute of Technology
Kyoto, Japan, July 10–14, 2000

National Aeronautics and
Space Administration

Glenn Research Center

Acknowledgments

The first author gratefully acknowledges the support of this work by the HPCC Program at NASA Glenn Research Center through Contract NAS3-97186.

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076
Price Code: A03

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100
Price Code: A03

Available electronically at <http://gltrs.grc.nasa.gov/GLTRS>

Parallel CE/SE Computations via Domain Decomposition

Ananda Himansu and Xiao-Yen Wang
Taitech, Inc
Cleveland, Ohio 44135

Philip C.E. Jorgenson and Sin-Chung Chang
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Abstract. This paper describes the parallelization strategy and achieved parallel efficiency of an explicit time-marching algorithm for solving conservation laws. The Space-Time Conservation Element and Solution Element (CE/SE) algorithm for solving the 2D and 3D Euler equations is parallelized with the aid of domain decomposition. The parallel efficiency of the resultant algorithm on a Silicon Graphics Origin 2000 parallel computer is checked.

1 Introduction

To accurately resolve complex three-dimensional turbulent flows, with possible acoustic waves or combustion phenomena, a time-dependent Euler or N-S solver must be used in Large Eddy Simulation or Direct Numerical Simulation mode. The need to resolve multiple space and time scales leads to very large computer memory and operation count requirements. The use of such computational resources is economically feasible only on modern parallel computing architectures. This paper presents the strategy and specific tools that were used to parallelize an existing serial code for solving the 3D Euler equations. While the concepts involved have been explored by other researchers over many years, the present authors hope that the description of a recent simple parallelization effort will be of value to many who are currently switching from serial supercomputers to more economical parallel architectures but who are not experts in parallel computing.

2 The CE/SE Euler Scheme

The Space-Time Conservation Element and Solution Element (CE/SE) method advanced by Chang [1] offers a simple numerical scheme that accurately resolves both shock waves and acoustic waves in the nearfield in a unified manner. The CE/SE 2D Euler solver is described in [2], while the CE/SE 3D Euler solver was presented in [3]. See the web site <http://www.lerc.nasa.gov/WWW/microbus> for details of the CE/SE method and a complete list of publications, including those detailing application of CE/SE schemes to computational aeroacoustics and combustion.

The success of the CE/SE technique for aeroacoustics, combustion, and other computations can be traced to the distinguishing features of the method, which

are (i) an emphasis on solving a general space-time integral form of the conservation laws, (ii) the requirement of local and global conservation of space-time fluxes as a basis for the method, (iii) use of only the simplest discretization stencils on a staggered space-time mesh, and (iv) the treatment of flow property gradients as additional unknowns. The basic inviscid flow solvers are nondissipative (no inherent artificial viscosity) and neutrally stable. A specifiable amount of artificial viscosity is explicitly added to provide stabilizing damping and to minimize numerical oscillations in the vicinity of shock waves. Thus the CE/SE Method is well suited to aeroacoustics and combustion applications.

3 Code Parallelization via Domain Decomposition

The CE/SE Euler scheme is an explicit time-marching scheme, making it well-suited for parallelization via domain decomposition. If the solution is to be computed at a time level n , the computational stencil involves a single cell at the time level n , and involves only the cell and its immediate neighbors at the time level $n - 1$. The algorithm to compute the solution is the same for every interior cell. Therefore, the scheme possesses a data parallelism. The only dependency in this data parallelism is that the solution and geometric information pertaining to immediate neighbors at the time level $n - 1$ needs to be known by any cell before the solution at that cell can be computed at the level n . The natural technique for parallelization is by dividing the cells among the available processors. Given the low speed of communication between processors relative to the speed with which a processor accesses its local memory, the division should be such that communication between processors is minimized.

To enable efficient parallelization of the CE/SE codes, a suitable domain decomposition strategy for the numerical flow problem has been selected. This involves use of one of the METIS codes based on algorithms described in [4]. Specifically, the dual-mesh partitioning code named *partdmesh* is used. It is available at no cost, and can be downloaded from the website <http://www-users.cs.umn.edu/~karypis/metis/main.shtml>. Given an unstructured mesh as a map of cell numbers to node (cell vertex) numbers, the METIS mesh-partitioning code produces a k -way partition of the domain, where the input k is the desired number of subdomains. Generally, k is set equal to the number of computer processors used for the flow computation. The METIS code can partition both 2D meshes of triangular cells and 3D meshes of tetrahedral cells. The partitioning algorithm attempts to produce an optimal partition in three respects. The algorithm attempts to balance the load, i.e., to assign an equal number of cells to each subdomain. The algorithm also attempts to minimize the number of shared cell faces at the inter-subdomain boundaries, so as to lessen the amount of inter-processor communication required. Thirdly, the algorithm also attempts to minimize the maximum of the number of subdomains adjacent to any subdomain. The METIS code is based on a multilevel graph-partitioning algorithm, and is very rapid, producing a 32-way partition of a 96800-cell domain in about 2 seconds on a personal computer with a 450 MHz Pentium-II processor. In

the current work, only the serial computer version of the METIS code is used, although a parallel computer version is also available. Other mesh-partitioning algorithms are available, but have not been evaluated for this work. In our experience, the METIS code produces partitions that are close to optimal, at very little computational cost.

The effort reported here consisted of “parallelizing” already existing 2D and 3D research codes that were written to run in serial fashion on a single processor. The parallelization was done in such a way as to introduce minimal modification of the existing flow solvers. The chief aim was to reduce the wallclock time of any given computation by using multiple processors. With regard to the actual flow computation, an additional benefit of the domain decomposition strategy is that the overall computer memory requirement is divided among the processors. However, little attention was paid to dividing among the processors the overall memory requirement of the domain decomposition and subdomain setup portions of the procedure. To enable the codes to be run on both parallel computers and workstation clusters, a message-passing approach was taken.

The CE/SE scheme is formulated on an unstructured mesh of triangles in the planar or axisymmetric case, and on an unstructured mesh of tetrahedra in the three-dimensional case. For simplicity of presentation, the case of a planar mesh of triangles will be discussed, though the parallelization procedure for volume meshes of tetrahedra is similar. The output typical of an unstructured mesh generator consists of the number of nodes, V , the number of cells, C , the number of boundary cell faces, B , the spatial coordinates X of each node, the map $C2V$ of cell number to the node numbers of its vertices (i.e., the cell definitions), a map $B2T$ of the boundary face number to a label denoting the type of boundary condition, the map $B2V$ of the boundary face number to the node numbers of its vertices, and the map $B2C$ of the boundary face number to the cell number of the cell to which the boundary face belongs. If the cell c has vertices with node numbers v_1 , v_2 and v_3 , then $C2V(c, i) = v_i$, $i = 1, 2, 3$. The map $C2V$ is supplied as input to the METIS mesh partitioning code together with the total number P of processors or subdomains desired. The output from the METIS code is a map $C2P$ of cell number to processor number, where the processor number to which the cell is assigned lies in the range 0 through $P - 1$.

A simple example of a planar mesh is considered in Fig. 1. The mesh happens to be structured. However, structured meshes are treated in exactly the same way as a general unstructured mesh. The mesh of triangles covers the domain $ABCDEFGHA$. The total number of cells here is $C = 8$. The nodes are labeled A through I in the figure, to avoid confusion with the cell numbers. Suppose that the mesh is partitioned into two subdomains, with cells 1 through 4 being assigned to Processor 0, and cells 5 through 8 to Processor 1. The current practice in the CE/SE schemes is to implement boundary conditions by adjoining “ghost” cells of appropriate geometry outside the domain of the flow problem. These ghost cells, one at each boundary face, are numbered -1 through $-B$, see Fig. 1. A ghost cell at any boundary face that does not call for a periodicity type of boundary condition is assigned to the same processor as the interior cell

with which it shares the boundary face. A ghost cell at a periodic boundary face is given the same geometric shape and assigned the same processor as the interior cell at the periodic image of the boundary face. For example, in Fig. 1, if the faces AB and GF are locations of periodicity of the solution, and if cell 2 is assigned to processor 0 while cell 5 is assigned to processor 1, then cell -6 is the periodic image of cell 2 and is assigned to processor 0, while cell -1 is the periodic image of cell 5 and is assigned to processor 1.

A face numbering is next generated, and the face-to-cell map is calculated. For very large meshes, this process may require too much memory to be done on a single processor, and a less memory-intensive approach would be needed to generate equivalent information. The cells marked for each processor p are given cell numbers “local” to the processor. Thus, the local numbering of the interior cells of processor p runs from 1 through C_p , and that of the ghost cells of processor p runs from -1 through $-B_p$. Local node numbering of the relevant nodes is also performed. The subdomain setup code then scans the face-to-cell map to find faces that are shared by processor p with a different processor. If there are D_p such faces, the cell at each face that is “foreign” to the processor p is assigned a local cell number in the range $C_p + 1$ through $C_p + D_p$, in the order in which the face is encountered in the face-to-cell map. The number of neighboring processors of any processor p is counted during this scan, and for each neighboring processor, the cell numbers (local to processor p) of both the “native” cells and the “foreign” cells are stored in the order in which the face is encountered in the face-to-cell map. An important property of this algorithm is that faces shared by two given processors p and p' are encountered in the same order when considering either p or p' . During the identification and local numbering of cells relevant to any processor p , maps from the local to global cell and node numberings are constructed, together with the inverse maps. Finally, for each processor p , mesh information is translated from global to local numbering and written to a disk file (with filename keyed to the processor number) about all the native interior and ghost cells, and about all the “neighboring” cells at the shared border of the subdomain. The “neighboring” cells appear merely as passive cells on processor p . They are used to store the flow data transferred from the neighboring processors. For each processor, the list of neighboring processors and the lists of native and foreign cells at the border with each processor are also written to disk files.

For a k -way partition of the domain, a CE/SE program executable is loaded into the node memory of each of k processors. Each of these processes then computes the solution in the corresponding subdomain. Only the cells assigned to each processor, together with passive border cells belonging to neighboring processors are stored, and the numbering is local. After each time step, the solution at the passive border cells is updated. This is done by message passing between processors, using Message Passing Interface (MPI) library calls. See [5] and the web site <http://www-unix.mcs.anl.gov/mpi> for details of the MPI standard. If necessary, the solution on the complete domain can be stitched together from the solutions on the individual subdomains by postprocessing.

4 Achieved Parallel Speedup

The parallel efficiency of the 3D Euler code when running on an SGI Origin 2000 computer architecture has been measured. The Origin 2000 uses a hypercube network topology to maximize the communication bandwidth. The parallel speedup achieved on this architecture is shown in Figs. 2 and 3. The test involved computing 100 time steps on a mesh of 172800 cells, using 1, 2, 4, 8, 16, 32 and 64 processors. The test problem was the reflection of an oblique shock wave from a wall, previously computed using the 2D Euler code in [2], but computed here using the 3D code. The parallel speedup of a run is defined as the inverse ratio of the wallclock time to the wallclock time of the single-processor case. The `MPI_WTIME` function call was used to reliably recover the wallclock time for each processor. The wallclock time of a run was taken to be the maximum time among the processors. The parallel speedup is seen to deviate visibly from ideal in the 32 processor case, indicating that for this implementation on this computer architecture, about 5400 mesh cells per processor is the limit below which the higher ratio of communication to computation will give poor parallel efficiency. Using 64 processors for this mesh size (about 2700 cells per processor) leads to a parallel speedup of only about 48. The same flow problem was run on a mesh four times as large, i.e. 691200 cells, with 64, 128 and 256 processors. This mesh was too large to be run on a single processor. Therefore, the 64 processor case was treated as having ideal speedup of 64, in order to examine the rate of falloff of the speedup, plotted in Fig. 3. Comparison of Figs. 2 and 3 shows that the rate of falloff (i.e., the slope) is approximately the same at 64 processors in Fig. 2 as it is at 256 processors in Fig. 3.

5 Summary and Conclusions

The CE/SE 2D and 3D Euler schemes were extended from serial computer versions to parallel computer versions. The parallel speedup of the 3D code was significantly less than ideal when an increased number of processors brought the number of mesh cells per processor below 5400. This indicates that tuning of the interprocessor communication will be beneficial when using a large number of processors on moderate sized meshes.

References

1. S.C. Chang, "The Method of Space-Time Conservation Element and Solution Element - A New Approach for Solving the Navier-Stokes and Euler Equations", *J. Comput. Phys.*, 119, pp. 295-324, (1995).

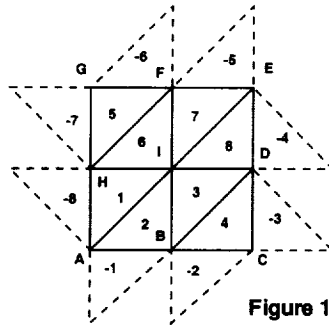


Figure 1

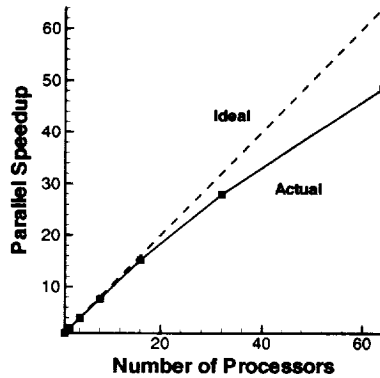


Figure 2

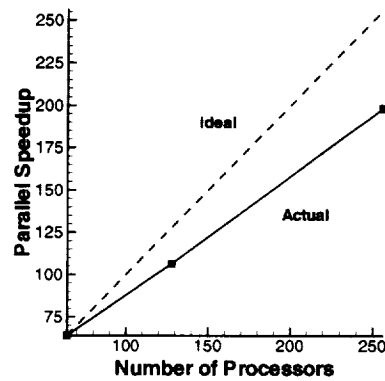


Figure 3

2. S.C. Chang, X.Y. Wang and C.Y. Chow, "The Space-Time Conservation Element and Solution Element Method : A New High-Resolution and Genuinely Multidimensional Paradigm for Solving Conservation Laws", J. Comput. Phys., 156 (1999), pp. 89-136.
3. X.Y. Wang and S.C. Chang, "A 3D Structured/Unstructured Euler Solver based on the Space-Time Conservation Element and Solution Element Method", 14th AIAA CFD Conference, June 1999, Norfolk, Virginia.
4. G. Karypis and V. Kumar, "Multilevel k-way Partitioning Scheme for Irregular Graphs", University of Minnesota, Dept. of Comp. Sc. / Army HPC Research Ctr., Technical Report 95-064, to appear in Journal of Parallel and Distributed Computing.
5. Message Passing Interface Forum, "MPI: A Message Passing Interface Standard", Intl. J. of Supercomputer Applications, vol. 8, no. 3/4, pp. 165-414, 1994. Available at <http://www.mpi-forum.org>.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE October 2000		3. REPORT TYPE AND DATES COVERED Technical Memorandum
4. TITLE AND SUBTITLE Parallel CE/SE Computations via Domain Decomposition			5. FUNDING NUMBERS WU-509-10-24-00	
6. AUTHOR(S) Ananda Himansu, Philip C.E. Jorgenson, Xiao Y. Wang, and Sin-Chung Chang				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration John H. Glenn Research Center at Lewis Field Cleveland, Ohio 44135-3191			8. PERFORMING ORGANIZATION REPORT NUMBER E-12476	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001			10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA TM-2000-210480	
11. SUPPLEMENTARY NOTES Prepared for the First International Conference on Computational Fluid Dynamics sponsored by the Kyoto Institute of Technology, Kyoto, Japan, July 10-14, 2000. Ananda Himansu and Xiao Y. Wang, Taitech Inc., Cleveland, Ohio 44135; Philip C.E. Jorgenson and Sin-Chung Chang, NASA Glenn Research Center. Responsible person, Philip C.E. Jorgenson, organization code 5880, 1-216-433-5386.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Categories: 02 and 64 This publication is available from the NASA Center for AeroSpace Information, 1-301-621-0390.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This paper describes the parallelization strategy and achieved parallel efficiency of an explicit time-marching algorithm for solving conservation laws. The Space-Time Conservation Element and Solution Element (CE/SE) algorithm for solving the 2D and 3D Euler equations is parallelized with the aid of domain decomposition. The parallel efficiency of the resultant algorithm on a Silicon Graphics Origin 2000 parallel computer is checked.				
14. SUBJECT TERMS Computational fluid dynamics; Flux conservation; Parallel processing; Domain decomposition			15. NUMBER OF PAGES 12	
			16. PRICE CODE A03	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	

